2. Moduláris programozás (UNIT-ok írása)

A **moduláris programozás** azt jelenti, hogy a programot részfeladatokra bontjuk, az egyes részeket külön programban dolgozzuk ki, ezeket a független modulokat lefordítjuk, és a végsõ programban összeépítjük õket. A Pascal moduljai a unitok: minden unit egy független csomag, melyben eljárások, függvények, típusok, konstansok és változók (a továbbiakban röviden: eljárások) vannak. Ha egy program deklarációs részében szerepel a USES unit deklaráció, a program felhasználhatja a unit eljárásait, mintha csak õbenne lennének deklarálva.

A Pascal elõre elkészített unitjai (Graph, Crt...) mellett mi is készíthetünk unitokat.

Saját unitokat két esetben készítünk. Lehetnek olyan eljárásaink, melyeket több programban is fel szeretnénk használni (pl. rajzoló segédeljárások). Ahelyett, hogy az egyik programból a vágólap segítségével átmásolgatnánk a többibe, ezeket az eljárásokat egy külön unitba gyûjtjük össze. Másrész, a unitok használata áttekinthetõbbé tesz egy nagyobb programot. Azzal, hogy a bizonyos feladatokat végzõ eljárásokat egy unitba gyûjtjük, az eljárások teszteléséhez nem kell az egész programot megírnunk, és a javításhoz elég csak a unitot átírni. A feladat modulokra bontásával az több programozó között is szétosztható, így lehetõvé válik a csoportos programozás.

Egy unit szerkezete:

UNIT unitnév;  
INTERFACE  
   közös rész  
IMPLEMENTATION  
   saját rész  
BEGIN  
   program  
END.

A közös részben szerepel mindaz, amit a unitot felhasználó programok használhatnak: változók, típusok, konstansok; de a közös eljárásoknak és függvényeknek csak a fejrésze. Maguk az eljárások és függvények csak az IMPLEMENTATION szó után következnek.

A saját részben szereplõ dolgok csak a unitra tartoznak: tehát itt szerepelnek pl. azok a változók, melyekre a unitnak szüksége van, de a unitot használó program számára láthatatlanok.

A program akkor hajtódik végre, amikor a unitot használó program elindul. Ezt a részt üresen is lehet hagyni.

A unit lefordítás után .TPU (tehát nem .EXE) kiterjesztésû file-ként kerül a lemezre (ebbõl következik, hogy használat elõtt a unitot lemezre kell fordítani). Nézzünk egy példát unit használatára:

UNIT Vonalak;  
INTERFACE  
   Procedure Vvonal(s:integer);  
   Procedure Fvonal(o:integer);  
IMPLEMENTATION  
   Uses Crt;  
   Procedure Vvonal(s:integer);  
    var i:integer;  
    Begin  
       for i:=1 to 80 do begin  
          gotoxy(i,s); write('-');  
       end;  
    End;  
   Procedure Fvonal(o:integer);  
    var i:integer;  
    Begin  
       for i:=1 to 25 do begin  
          gotoxy(o,i); write('|');  
       end;  
    End;  
  
BEGIN  
   clrscr;  
END.

Ezt a unitot VONALAK.PAS néven kell lemezre menteni, majd lemezre fordítani. Figyeljük meg, hogy a saját részben az eljárások fejét megismételtük. Ezt rövidebben is lehetett volna: mivel az eljárások teljes fejrésze már szerepel a közös részben, a saját részbe elég lett volna  
   Procedure Vvonal;  
   Procedure Fvonal;  
rövidített fejrészt írni.

A unitot használó program a következõképpen nézhet ki:

USES Vonalak;  
BEGIN  
 Vvonal(12);  
 Fvonal(40);  
END.

A program indításakor a képernyõ letörlõdik. Megjegyzés: a program nem használhatja a Crt unit eljárásait, hiszen az nincs deklarálva benne.